We Claim:

1. A method for applying a lens correction to image data that is associated with a lens, the method comprising:

converting the image data to a YUV color space to form YUV image data, if the image data is not in the YUV color space;

applying image processing procedures to the YUV image data to form image processed YUV data; and

applying the lens correction to the image processed YUV data.

2. The method of Claim 1, wherein applying the lens correction further comprises:

applying a Y correction value to a Y component of the image processed YUV data;

applying a U correction value to a U component of the image processed YUV data; and

applying a V correction value to a V component of the image processed YUV data.

3. The method of Claim 1, wherein applying the lens correction further comprises:

multiplying a Y component of the image processed YUV data by a Y correction value;

adding a U component of the image processed YUV data to a U correction value; and

adding a V component of the image processed YUV data to a V correction value.

4. The method of Claim 2, wherein the U correction value is based on a first distance value, wherein the first distance value is associated with a location of a target pixel in a reference image from a reference point of the reference image.

5. The method of Claim 2, wherein the U correction value is based on a luminance parameter, wherein the luminance parameter is determined based on whether the Y component of the image processed YUV data falls within a pre-selected luminance range..

6. The method of Claim 2, wherein the U correction value is based on a maximum correction limit and a minimum correction limit.

7. The method of Claim 6, wherein the maximum correction limit and the minimum correction limit are user-selected.

8. The method of Claim 6, wherein the maximum correction limit and the minimum correction limit are based on properties of the lens.

9. The method of Claim 4, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rings emanating from the reference point of the reference image.

10. The method of Claim 4, wherein the first distance value is calculated by

first distance value = Root (Dx*Dx + Dy*Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

11. The method of Claim 4, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rectangles emanating from the reference point of the reference image.

12. The method of Claim 4, wherein the first distance value is calculated by

first distance value = max (Dx,Dy)* NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

13. The method of Claim 4, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rhombuses emanating from the reference point of the reference image.

14. The method of Claim 4, wherein the first distance value is calculated by

first distance value) = (Dx + Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

15. The method of Claim 4, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric polygons emanating from the reference point of the reference image, wherein the plurality of concentric polygons are substantially ring-shaped.

16. The method of Claim 4, wherein the first distance value is calculated by

If (Dx > (Dy<<2))

then,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy) + (abs(Dx –(Dy<<2))>>3))) * NormalizeValue

Else If (Dy > (Dx<<2))

then,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy)+ (abs(Dy –(Dx<<2))>>3)) * NormalizeValue

Else If (max(Dx,Dy) > (abs(Dx-Dy)<<2))

then,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy) + (max(Dx,Dy) – (abs(Dx-Dy)<<2)>>3)) * NormalizeValue

Else,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy)) * NormalizeValue

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

17. The method of Claim 2, wherein the V correction value is based on a first distance value, wherein the first distance value is associated with a location of a target pixel in a reference image from a reference point of the reference image.

18. The method of Claim 2, wherein the V correction value is based on a luminance parameter, wherein the luminance parameter is determined based on whether the Y component of the image processed YUV data falls within a pre-selected luminance range.

19. The method of Claim 2, wherein the V correction value is based on a maximum correction limit and a minimum correction limit.

20. The method of Claim 19, wherein the maximum correction limit and the minimum correction limit are user-selected.

21. The method of Claim 19, wherein the maximum correction limit and the minimum correction limit are based on properties of the lens.

22. The method of Claim 17, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rings emanating from the reference point of the reference image.

23. The method of Claim 17, wherein the first distance value is calculated by

first distance value = Root (Dx*Dx + Dy*Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

24. The method of Claim 17, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rectangles emanating from the reference point of the reference image.

25. The method of Claim 17, wherein the first distance value is calculated by

first distance value = max (Dx,Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

26. The method of Claim 17, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rhombuses emanating from the reference point of the reference image.

27. The method of Claim 17, wherein the first distance value is calculated by

first distance value = (Dx + Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

28. The method of Claim 17, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric polygons emanating from the reference point of the reference image, wherein the plurality of concentric polygons are substantially ring-shaped.

29. The method of Claim 17, wherein the first distance value is calculated by

If (Dx > (Dy<<2))

then,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy) + (abs(Dx –(Dy<<2))>>3))) * NormalizeValue

Else If (Dy > (Dx<<2))

then,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy)+ (abs(Dy –(Dx<<2))>>3)) * NormalizeValue

Else If (max(Dx,Dy) > (abs(Dx-Dy)<<2))

then,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy) + (max(Dx,Dy) – (abs(Dx-Dy)<<2)>>3)) * NormalizeValue

Else,

Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy)) * NormalizeValue

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

30. The method of Claim 2, wherein the Y correction value is based on a second distance value, wherein the second distance value is in turn based on a first distance and one or more luminance parameters based on an F value of the lens.

31. The method of Claim 2, wherein the Y correction value is based on a smoothing parameter, wherein the smoothing parameter is user-selected based on a desired amount of smoothing.

32. The method of Claim 2, wherein the Y correction value is based on a maximum correction limit and a minimum correction limit.

33. The method of Claim 32, wherein the maximum correction limit and the minimum correction limit are user-selected.

34. The method of Claim 32, wherein the maximum correction limit and the minimum correction limit are based on properties of the lens.

35. The method of Claim 30, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rings emanating from the reference point of the reference image.

36. The method of Claim 30, wherein the first distance value is calculated by

first distance value = Root (Dx*Dx + Dy*Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

37. The method of Claim 30, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rectangles emanating from the reference point of the reference image.

38. The method of Claim 30, wherein the first distance value is calculated by

first distance value = max (Dx,Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

39. The method of Claim 30, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric rhombuses emanating from the reference point of the reference image.

40. The method of Claim 30, wherein the first distance value is calculated by

first distance value = (Dx + Dy) * NormalizeValue,

wherein:

Dx = abs(HalfX – x + XSHIFT);

Dy = abs(HalfY – y + YSHIFT);

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.

41. The method of Claim 30, wherein the first distance value is calculated by assuming that target pixels in the reference image lie in a plurality of concentric

polygons emanating from the reference point of the reference image, wherein the plurality of concentric polygons are substantially ring-shaped.

42. The method of Claim 30, wherein the first distance value is calculated by

```
If (Dx > (Dy<<2))
then,
Function_Distance(x,y) = ( Dx + Dy +max(Dx , Dy) + (abs(Dx –(Dy<<2))>>3))) *
        NormalizeValue
Else If (Dy >  (Dx<<2))
then,
Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy)+ (abs(Dy –(Dx<<2))>>3)) *
        NormalizeValue
Else If (max(Dx,Dy) > (abs(Dx-Dy)<<2))
then,
Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy) + (max(Dx,Dy) – (abs(Dx-
        Dy)<<2)>>3) ) * NormalizeValue
Else,
Function_Distance(x,y) = (Dx + Dy +max(Dx , Dy) ) * NormalizeValue
wherein:
        Dx = abs(HalfX – x + XSHIFT);
        Dy = abs(HalfY – y + YSHIFT);
```

HalfX is half a length of the reference image in an x direction; and

HalfY is half a width of the reference image in a y direction.